

Session Parameters

Natural session parameters may be used with certain Natural statements to control such factors as the size of a report, how fields are to be displayed, etc.

This section covers the following topics:

- General Information
 - How to Set Parameters
 - Evaluation of Parameters
 - Overview of Parameters
-

General Information

In Natural, several parameters are used:

- to specify certain characters,
- to set processing time limits,
- to set a particular response for a given condition,
- to set various size limits,
- to determine various aspects of output reports.

The statements in which a session parameter may be set/evaluated are indicated in the section Overview of Parameters as well as in the description of each parameter.

At the installation of Natural, the Natural administrator sets these parameters to default values which are then valid for all users of Natural.

To see which parameter values apply to your session, you enter the system command GLOBALS as described in the Natural User's Guide.

How to Set Parameters

Natural session parameters can be set in several ways:

- via the default parameter module/file (NATPARM), which is set when Natural is installed;
- via dynamic parameters specified when invoking Natural (as described in your Natural Operations documentation);
- via the system command GLOBALS;
- via a SET GLOBALS statement (in reporting mode only);
- via a FORMAT statement;
- via parameter specification within statements where parameters also are evaluated, for example, INPUT, DISPLAY, WRITE;
- via terminal commands.

Instead of the parameter values "ON" and "OFF", you can also specify "T" (true) or "F" (false) respectively.

Changing Parameters at Session Level - The GLOBALS Command

For your Natural session you can change some of the parameter values set by the Natural administrator.

Within your Natural session, you can change these parameters by issuing the following system command:

GLOBALS

When you issue the GLOBALS command, a screen is displayed showing the parameter values that are currently in effect for your session. On this screen, you can change the values that do not suit your requirements.

A parameter value set with a GLOBALS command remains in effect until the end of the Natural session (and applies to every object you store during the session), unless you change it again with another GLOBALS command.

Changing Parameters at Program Level - The FORMAT Statement

You can change certain parameters for the duration of a single program (report). This is done by using a FORMAT statement in the program, which will override the session-wide settings for these parameters.

Example of a FORMAT Statement:

```
FORMAT AL=10 HC=R
```

Parameters set with a FORMAT statement apply until the end of the executed program, unless they are changed with another FORMAT statement in the program.

Not all session parameters can be changed at program level, while several parameters that can be specified at program level cannot be specified at session level; most of the latter are parameters which affect the format of an output report.

Changing Parameters at Statement Level

Most of the parameters you can change with a FORMAT statement you can also change for an individual statement; for example, for a particular DISPLAY, WRITE, INPUT or REINPUT statement.

This is done by specifying the parameter (in parentheses) after the statement name.

Example:

```
DISPLAY (SF=4) NAME JOB-TITLE CURR-CODE SALARY
```

A parameter set at statement level applies only to the statement in which it is specified. The setting at statement level overrides, for that statement only, all other settings of that parameter at other levels.

Changing Parameters at Field Level

Within a DISPLAY, WRITE, INPUT or REINPUT statement, you can also change some parameters for an individual field or output element.

This is done by specifying the parameter (in parentheses) after the field name.

Example:

```
DISPLAY NAME (AL=10) JOB-TITLE CURR-CODE SALARY
```

The parameter value then applies only to that field. The setting at field level overrides, for that field only, all other settings of that parameter at other levels. However, only some of the parameters that can be set at statement level can also be set at field level.

Evaluation of Parameters

Parameters specified with the statements DISPLAY, FORMAT, PRINT, INPUT, REINPUT, WRITE, WRITE TITLE and WRITE TRAILER are processed during program compilation and are therefore included in the corresponding object module for the program.

The following hierarchy is used for evaluation:

1. Parameters set at element/field (highest priority)
2. Parameters set at statement level
3. Parameters set with a FORMAT statement
4. The default parameter settings (lowest priority)

Parameters set with a SET GLOBALS statement cause the execution time environment to be modified. These modifications remain in effect until overridden by another SET GLOBALS statement (or GLOBALS command).

Overview of Parameters

X - Parameter may be specified dynamically / with GLOBALS command / with FORMAT statement.

S - Parameter may be specified at Statement level.

E - Parameter may be specified at Element level.

C - Parameter is evaluated at Compilation time.

R - Parameter is evaluated at Runtime.

	Default Value	Dynamic Parameter	GLOBALS Command	Statements						
				SET GLOBALS	FORMAT	DISPLAY	INPUT	REINPUT	WRITE	PRINT
AD - Attribute Definition					X	SE	SE	SE	SE	SE
AL - Alphanumeric Length for Output	none				X	SE	SE		SE	SE
BX - Box Definition	none				X	SE	SE		SE	
CC - Conditional Program Execution	OFF	X	X	R						
CD - Color Definition	NE				X	SE	SE	SE	SE	SE
CF - Character for Terminal Commands	%	X	X	R						
CO - Compiler Output	OFF	X	X							
CV - Control Variable	none					SE	SE		SE	SE
DC - Character for Decimal Point Notation	.	X	X	R						
DF - Date Format	S				X	SE	SE		SE	SE

	Default Value	Dynamic Parameter	GLOBALS Command	Statements						
				SET GLOBALS	FORMAT	DISPLAY	INPUT	REINPUT	WRITE	PRINT
DFOUT - Date Format for Output	S	X	X	R						
DFSTACK - Date Format for Stack	S	X	X	R						
DFTITLE - Date Format in Default Page Title	S	X	X	R						
DU - Dump Generation	OFF	X	X	R						
DY - Dynamic Attributes	none					SE	SE		SE	SE
EJ - Page Eject	ON	X	X	R						
EM - Edit Mask	none				X	SE	SE		SE	SE
ES - Empty Line Suppression	OFF				X	S			S	
FC - Filler Character	blank				X	SE				
FCDP - Filler Character for Dynamically Protected Fields	ON	X	X	R						
FL - Floating Point Mantissa Length	none				X	SE	SE		SE	SE
FS - Format Specification	OFF	X	X	R						
GC - Filler Character for Group Headers	blank				X	SE				
HC - Header Centering	C				X	SE				
HE - Help routine	none						SE			
HW - Heading Width	ON				X	S				

	Default Value	Dynamic Parameter	GLOBALS Command	Statements						
				SET GLOBALS	FORMAT	DISPLAY	INPUT	REINPUT	WRITE	PRINT
IA - INPUT Assign Character	=	X	X	R						
IC - Insertion Character	none				X	SE				
ID - INPUT Delimiter Character	,	X	X	R						
IM - INPUT Mode	F	X	X	R						
IP - INPUT Prompting Text	ON				X		SE			
IS - Identical Suppress	OFF				X	SE			SE	
KD - Key Definition	OFF				X					
LC - Leading Characters	none				X	SE				
LE - Limit Error Processing	OFF	X	X	C						
LS - Line Size	physical	X	X	C	X	S	S		S	
LT - Limit of Records Read	99999999	X	X	R						
MC - Multiple-Value Field Count	1				X	S	S		S	S
MP - Maximum Number of Pages	32767				X	S			S	S
MS - Manual Skip	OFF				X		S			
MT - Maximum CPU Time	60 sec.	X	X	R						
NC - Use of Natural System Commands	OFF	X	X	R						

	Default Value	Dynamic Parameter	GLOBALS Command	Statements						
				SET GLOBALS	FORMAT	DISPLAY	INPUT	REINPUT	WRITE	PRINT
NL - Numeric Length for Output	none				X	SE	SE		SE	SE
OPF - Overwriting of Protected Fields by Help routines	ON	X	X	R						
PC - Periodic Group Count	1				X	S	S		S	S
PD - NATPAGE Page Dataset	50	X	X	R						
PM - Print Mode	none	X	X	C	X	SE	SE		SE	SE
PS - Page Size	physical	X	X	CR	X	S	S		S	
REINP - Internal REINPUT for Invalid Data	ON	X	X	R						
SA - Sound Terminal Alarm	OFF	X	X	R						
SF - Spacing Factor	1	X	X	C	X	SE				
SG - Sign Position	ON				X	SE	SE		SE	SE
SL - Source Line Length	72	X	X							
SM - Structured Mode	OFF	X	X							
TC - Trailing Characters	none				X	SE				
TS - Translate Output from Programs in System Libraries	OFF	X	X							
UC - Underlining Character	-				X	SE				

	Default Value	Dynamic Parameter	GLOBALS Command	Statements						
				SET GLOBALS	FORMAT	DISPLAY	INPUT	REINPUT	WRITE	PRINT
WH - Wait for Record in Hold Status	OFF	X	X	R						
ZD - Zero Division Check	ON	X	X	R						
ZP - Zero Printing	ON	X	X	C	X	SE	SE	SE	SE	SE

AD - Attribute Definition

Parameter	Possible Values	Default Value	Applicable Statements
AD	see below	see below	CALLNAT DISPLAY FORMAT INPUT PERFORM PRINT REINPUT WRITE

With this parameter, you can specify field attributes at field or statement level.

You can specify multiple attributes in any sequence:

AD =	$ \begin{array}{c} \text{B} \\ \text{C} \\ \text{D} \\ \text{I} \\ \text{N} \\ \text{U} \\ \text{V} \\ \text{Y} \end{array} \begin{array}{c} \text{L} \\ \text{R} \\ \text{Z} \end{array} \begin{array}{c} \text{A} \\ \text{M} \\ \text{O} \\ \text{P} \end{array} \begin{array}{c} \text{E} \\ \text{F} \end{array} \begin{array}{c} \text{G} \\ \text{H} \end{array} \begin{array}{c} \text{T} \\ \text{W} \end{array} [\text{c}'] $
------	---

The meaning of each value is explained on the following pages.

Default values are underlined.

For the alignment of field values (2nd column of values in above syntax), the default value is

L - for alphanumeric fields, and

R - for numeric fields.

Examples:

```
DISPLAY #FIELD A (AD=R) INPUT #FIELD B (AD=M) INPUT (AD=IM) #FIELD A #FIELD B
```

Field Representation

Value	Meaning	Statements	Explanation
B	blinking (*)	all	The value of the field is displayed blinking.
C	cursive/italic (*)	all	The value of the field is displayed cursive/italic.
D	default intensity	all	The value of the field is displayed with normal intensity, that is, not highlighted in any way.
I	intensified	all	The value of the field is displayed intensified.
N	non-display	all	A value entered in the field will not be displayed.
U	underlined	all	The value of the field is displayed underlined.
V	reverse video (*)	all	The value of the field is displayed reverse video.
Y	dynamic attributes	INPUT DISPLAY WRITE	Attributes are to be controlled via a control variable (format C).

* The field representation attributes marked with an asterisk (*) require corresponding hardware features, and will be ignored at runtime if these features are not available.

Field Alignment

Value	Meaning	Statements	Explanation
L	left-justified	all	The value of the field is displayed left-justified.
R	right-justified	all	The value of the field is displayed right-justified.
Z	leading zeros	all	Numeric values are displayed with leading zeros, right-justified.

Field Input/Output Characteristics

Value	Meaning	Statements	Explanation
A	input field, non-protected	INPUT	The value of the field is to be entered in response to the INPUT statement.
M	output field, modifiable	INPUT CALLNAT	The value of the field is to be displayed during INPUT statement execution, and a different value may be entered by the user. The field is an output field and may be modified.
O	output field, write-protected	INPUT CALLNAT	The value of the field is to be displayed during INPUT execution. The field is an output field and may not be modified.
P	temporarily protected	INPUT REINPUT	Used in conjunction with control variable (format C), the DY parameter, and the REINPUT statement.

Mandatory Input

Value	Meaning	Statements	Explanation
E	value mandatory	INPUT	A value must be entered in the field in response to an INPUT statement; otherwise an error message will be issued. This is only relevant for input-only fields (AD=A).
F	value optional	INPUT	A value can, but need not, be entered in the field in response to an INPUT statement.

Length of Input Value

Value	Meaning	Statements	Explanation
G	value size	INPUT	The value entered in the field in response to an INPUT statement must be of the same length as the field. This is only relevant for input-only fields (AD=A).
H	value size	INPUT	The value entered in the field in response to an INPUT statement may be shorter than the field.

Field Upper/Lower Case Characteristics

Value	Meaning	Statements	Explanation
T	translate lower to upper case	INPUT	The value entered is to be translated to upper case.
W	accept lower case	INPUT	Lower case values are to be accepted.

Filler Character

Value	Meaning	Statements	Explanation
'c'	filler character	INPUT	The empty field is to be filled with the specified character c (for display only) if AD=A or AD=M is specified.

Note for mainframe computers:

If the filler character is set to blank (X'40'), filling blanks are replaced by X'00' to allow for insertion of characters without having to clear the remainder of the input field before. In BS2000 environments, X'00' characters are displayed as dots on 97xx type terminals. Their appearance can be changed by means of the SIDA utility or with the configuration utility of the respective terminal emulation.

AL - Alphanumeric Length for Output

Parameter	Possible Values	Default Value	Applicable Statements
AL	1 to n (n = value of LS parameter minus 1)	none	DISPLAY FORMAT INPUT PRINT WRITE

With this parameter, you specify the default output length for an alphanumeric field; that is, when it is specified shorter than the field length, the field will be right-truncated.

It is not recommended to use the AL parameter for input fields (AD=A or AD=M) in an INPUT statement.

Any edit mask specified for a field will override the AL parameter for this field.

Example:

```
FORMAT AL=20
```

BX - Box Definition

Parameter	Possible Values	Default Value	Applicable Statements
BX	T, B, L, R, ON, OFF	none	DISPLAY FORMAT INPUT WRITE

This parameter only applies to Natural on mainframe computers.

The outlining feature is only available on certain types of terminals, usually those which also support the display of double-byte character sets. If the terminal used does not support outlining, this parameter will be ignored at execution time.

Outlining (boxing) is the capability to generate a line around certain fields when they are displayed on the terminal screen. Drawing such "boxes" around fields is another method of showing the user the lengths of fields and their positions on the screen.

With this parameter, you specify which parts of a box are to be displayed:

T	Top horizontal line.
B	Bottom horizontal line.
L	Lefthand vertical line.
R	Righthand vertical line.

You can specify the values in any order.

Example:

DISPLAY #FIELD1 (BX=RLT) / #FIELD2 (BX=TLRB)
--

BX=ON corresponds to **BX=TBLR**.

BX=OFF causes no boxes to be drawn around the fields concerned.

See also the terminal command %D=.

CC - Conditional Program Execution

Parameter	Possible Values	Default Value	Applicable Statements
CC	ON / OFF	OFF	SET GLOBALS

This parameter only applies to Natural on mainframe computers.

With this parameter, you determine the action to be taken if an error is detected during compilation/execution of a Natural program in batch mode.

CC=ON	Natural will flush the input data stream for SYNIN and OBJIN until a line containing "%%" in the first two positions is encountered or until an end-of-file condition is detected. If more data are available in the input stream, Natural will resume reading after the line containing "%%".
CC=OFF	Natural will attempt to process the next program (or command) in the input stream.

When a Natural session terminates, return code 4 will be passed to the invoking program via register 15 if an error is detected (regardless of the CC setting).

Example:

```
SET GLOBALS CC=ON
```

CD - Color Definition

Parameter	Possible Values	Default Value	Applicable Statements
CD	BL = blue GR = green NE = neutral PI = pink RE = red TU = turquoise YE = yellow	NE	DISPLAY FORMAT INPUT REINPUT PRINT WRITE

With this parameter, you specify the color in which fields are to be displayed.

Note:

If no color screen is used, this parameter will be ignored at runtime.

Example:

```
INPUT (CD=RE) #A #B
```

CF - Character for Terminal Commands

Parameter	Possible Values	Default Value	Applicable Statements
CF	any special character, OFF	%	SET GLOBALS

With this parameter, you specify the control character for Natural terminal commands, that is, the character which is to be used as the first character of any terminal command.

For example, if the CF setting is "%" (which is the default), a terminal command must begin with the character "%".

If CF=OFF is specified, no control character for terminal commands will be available. Terminal commands issued with a SET CONTROL statement, however, will nonetheless be accepted. CF=OFF may only be specified dynamically when invoking Natural, or with a GLOBALS command, but not with a SET GLOBALS statement.

The character specified with the CF parameter must not be the same as the one specified with the HI parameter (help character) or IA parameter (input assign character).

Note:

The character specified with the CF parameter should not be the same as the one specified with the DC parameter (decimal character) or ID parameter (input delimiter character).

Example:

```
SET GLOBALS CF=+
```

Note:

In the map editor, the control character for terminal commands is always "%" (so as to avoid conflicts with delimiter characters used in maps), no matter which character is defined with the CF parameter.

CO - Compiler Output

Parameter	Possible Values	Default Value	Applicable Statements
CO	ON / OFF	OFF	GLOBALS command

This parameter should not be used without prior consultation of Software AG support. This parameter is not available on mainframe computers.

With this parameter, you determine whether a compiler listing is to be displayed when you compile a Natural object.

CO=ON	A compiler listing is displayed on the screen.
CO=OFF	No compiler listing will be displayed.

CV - Control Variable

Parameter	Possible Values	Default Value	Applicable Statements
CV	see below	none	DISPLAY INPUT PRINT WRITE

This parameter is used to reference a control variable.

A control variable is defined with format C and is used to assign field attributes dynamically.

The following field attributes can be modified dynamically:

- Field representation attributes
(B, C, D, I, N, U, V; see the session parameter AD).
- Field protection (P; see the session parameter AD).
- Color (BL, GR, NE, PI, RE, TU, YE; for an explanation of color codes, see the session parameter CD).

Example:

```
DEFINE DATA LOCAL 1 #ATTR(C) 1 #A (N5) END-DEFINE ... MOVE (AD=I CD=RE) TO #ATTR INPUT #A (CV=#ATTR) ...
```

A control variable can also be used to check whether the contents of fields have been modified during the execution of an INPUT statement (see also Modified Control Variables).

```
IF #ATTR MODIFIED ...
```

Note:

If the CV parameter is specified at statement level and at field level and the control variable for the individual field is empty, the control variable for the statement will be used for the field.

DC - Character for Decimal Point Notation

Parameter	Possible Values	Default Value	Applicable Statements
DC	any character (except numeric characters)	. (period)	SET GLOBALS

With this parameter, you specify the character to be used for decimal point notation.

The character assigned to DC will be in effect for all notations, where a decimal point is possible, i.e. variables, constants, edit masks.

The character specified with the DC parameter must not be the same as the one specified with the IA parameter (input assign character) or ID parameter (input delimiter character).

Note:

The character specified with the DC parameter should not be the same as the one specified with the CF parameter (control character for terminal commands) or HI parameter (help character).

Example:

```
SET GLOBALS DC=,
```

DF - Date Format

Parameter	Possible Values	Default Value	Applicable Statements
DF	S, I, L	S	COMPRESS DISPLAY FETCH FORMAT INPUT MOVE PRINT RUN STACK WRITE

When the value of a date field is converted to alphanumeric format (for example, in a MOVE, DISPLAY, WRITE or INPUT statement) and no edit mask is specified for the conversion, the default date format as determined by the profile parameter DTFORM is used as edit mask. The same is true for the input validation of a date variable used in an INPUT statement: If no edit mask is specified, the input is validated according to the date format determined by the DTFORM parameter.

With the DF parameter, you determine the length of a date when converted to alphanumeric representation without an edit mask being specified:

DF=S	8-byte representation with 2-digit year component and delimiters (yy-mm-dd).
DF=I	8-byte representation with 4-digit year component and no delimiters (yyyymmdd).
DF=L	10-byte representation with 4-digit year component and delimiters (yyyy-mm-dd).

Note:

The sequence of the day, month and year components and the delimiter characters used are determined by the DTFORM profile parameter (see your Natural Operations documentation).

With DF=S, only 2 digits are provided for the year information; this means that if the date value contained the century, this information would be lost during the conversion.

By using DF=I or DF=L, you can gradually change your applications to use 4-digit year representations and at the same time continue to make use of the flexibility provided by the DTFORM parameter.

The DF parameter is evaluated at compilation. It can be specified with the FORMAT statement, the statements INPUT, DISPLAY, WRITE and PRINT (at statement and field level), and the statements MOVE, COMPRESS, STACK, RUN and FETCH (at field level).

See also the section Processing of Date Information in the Natural Programming Guide.

DFOUT - Date Format for Output

Parameter	Possible Values	Default Value	Applicable Statements
DFOUT	S, I	S	SET GLOBALS

This parameter determines the format in which the values of date variables are displayed by INPUT, DISPLAY, PRINT and WRITE statements.

DFOUT=S	Date variables are displayed with a 2-digit year component, and delimiters as determined by the profile parameter DTFORM. For example: yy-mm-dd.
DFOUT=I	Date variables are displayed with a full 4-digit year component and no delimiters. For example: yyyyymmdd.

The DFOUT parameter is evaluated at runtime. It applies to date fields in INPUT, DISPLAY, PRINT and WRITE statements for which no explicit edit mask is specified and for which the session parameter DF is not set.

For either DFOUT setting, the sequence of the day, month and year components in the date values is determined by the profile parameter DTFORM (see your Natural Operations documentation).

See also the section Processing of Date Information in the Natural Programming Guide.

DFSTACK - Date Format for Stack

Parameter	Possible Values	Default Value	Applicable Statements
DFSTACK	S, C, I	S	SET GLOBALS

This parameter determines the format in which the values of date variables are placed on the stack via a STACK, RUN or FETCH statement.

DFSTACK=S	Date variables are placed on the stack with a 2-digit year component, and delimiters as determined by the profile parameter DTFORM. For example: yy-mm-dd.
DFSTACK=C	Same as DFSTACK=S. In addition, a change in the century (that is, if the century used when the value is read from the stack is not the same as the century of the original date value before it was stacked) will be intercepted at runtime.
DFSTACK=I	Date variables are placed on the stack with a full 4-digit year component and no delimiters. For example: yyyyymmdd.

The DFSTACK parameter does not apply to STACK, RUN and FETCH statements for which the session parameter DF is set.

See also the section Processing of Date Information in the Natural Programming Guide.

DFTITLE - Date Format in Default Page Title

Parameter	Possible Values	Default Value	Applicable Statements
DFTITLE	S, L, I	S	SET GLOBALS

This parameter determines the output format of the date in a default page title (as output with a DISPLAY, WRITE or PRINT statement).

DFTITLE=S	The date is output with a 2-digit year component and delimiters. For example: <i>yy-mm-dd</i> .
DFTITLE=L	The date is output with a 4-digit year component and delimiters. For example: <i>yyyy-mm-dd</i> .
DFTITLE=I	The date is output with a 4-digit year component and no delimiters. For example: <i>yyyymmdd</i> .

Note:

The sequence of the day, month and year components and the delimiter characters used are determined by the DTFORM profile parameter (see your Natural Operations documentation).

The DFTITLE parameter is evaluated at runtime. It has no effect on a user-defined page title (as specified with a WRITE TITLE statement).

See also the section Processing of Date Information in the Natural Programming Guide.

DU - Dump Generation

Parameter	Possible Values	Default Value	Applicable Statements
DU	ON, OFF, SNAP, FORCE	OFF	SET GLOBALS

DU on Mainframe Computers

With this parameter, you determine whether a memory dump is to be generated in the case of an abnormal interruption during the execution of a Natural program.

DU=OFF	No memory dump will be produced, error 954 ABEND at execution time will be produced, and subsequent action taken by Natural is determined by the setting of the CC parameter.
DU=ON	A memory dump will be produced in the case of an abnormal termination.
DU=SNAP	This will force an immediate dump in the case of an abnormal termination during a Natural session. The Natural session will continue after the dump has been taken.
DU=FORCE	This will force an immediate dump in the case of an abnormal termination during a Natural session and will terminate the Natural session immediately. This will be useful for testing purposes in some environments.

Example:

```
SET GLOBALS DU=ON
```

Notes:

Be careful when you use this parameter, because all programs and subroutines currently active for the current user will be retained in the Natural buffer pool.

DU=ON, SNAP or FORCE may cause buffer fragmentation which may result in a significant degradation in system performance.

Note for UTM:

Under UTM, this parameter is ignored; under UTM, a dump is always produced in the case of an abnormal program termination.

DU under OpenVMS, UNIX and Windows

With this parameter, you determine whether a disassembled object-code dump is to be generated.

DU=ON	<p>When one of the commands Windows CHECK, STOW, CATALOG or RUN is executed, a disassembled object-code file will be produced. This dump file will be written into the directory "tmp" below the Natural root directory.</p> <p>The name of the dump file consists of the source-file name and the extension ".DIA". If the source file has not been saved, the name of the dump file will be "GEN.DIA" (under OpenVMS and UNIX) or "UNTITLED.DIA" (under Windows) respectively.</p> <p>If the program contains database statements, dump files with the extension ".ADA" (for Adabas) or ".SQL" (for SQL databases) will also be generated.</p> <p>If cross-reference data are generated, a file with the extension ".XRF" will also be generated.</p>
DU=OFF	No dump will be produced.

Example:

```
SET GLOBALS DU=ON
```

Note:

DU=ON may create a large dump file (depending on the size of the source file), which can cause a significant degradation in system performance.

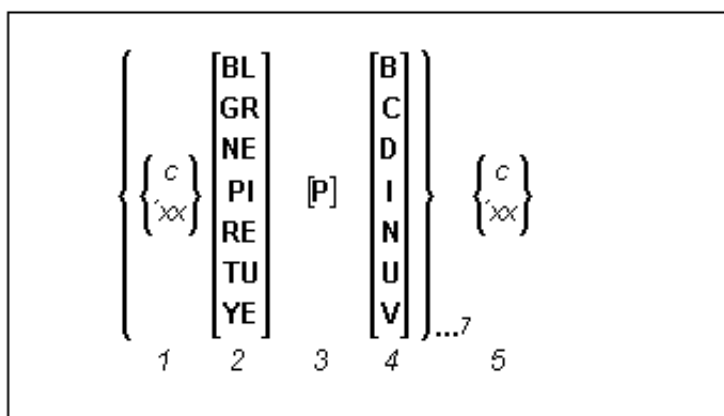
DY - Dynamic Attributes

Parameter	Possible Values	Default Value	Applicable Statements
DY	see below	none	DISPLAY INPUT PRINT WRITE

This parameter is used to assign attributes for dynamic attribute field display.

Special identification characters (escape characters) are used to indicate the beginning and end of attribute definitions.

An alphanumeric field which is processed with an INPUT, DISPLAY, WRITE or PRINT statement, and which contains escape characters, is split into subfields at the escape character position. The corresponding attribute is then assigned to the subfield. A blank is substituted for the escape character.



1. An escape character which denotes the beginning of the attribute definition. Any special character (or a hexadecimal number preceded by an apostrophe) may be used.
2. The color attribute to be assigned (for explanation of color codes, see the session parameter CD).
3. Subfield is to be write-protected.
4. Additional attributes to be assigned (see the session parameter AD).
5. An escape character which denotes the end of the attribute definition. Any special character (or a hexadecimal number preceded by an apostrophe) may be used.

You may specify up to seven escape sequences (escape characters and attributes) before the character indicating the end of the attribute definitions.

Note:

For a part of a field for which a DY specification takes effect, any specification made with a CV parameter will be ignored.

Example 1:

DY = <U>

The text string: THIS <is> UNDERLINED
is printed as: THIS is UNDERLINED

Example 2:

DY = <BL|RE/GR>

Assigns: Blue to "<"

Red to "|"

Green to "/"

">" switches back to the initial field color.

Example 3:

DY = <P>

The text string: **Do not overwrite <this>**

is printed as: **Do not overwrite this** (where "this" is protected)

EJ - Page Eject

Parameter	Possible Values	Default Value	Applicable Statements
EJ	ON / OFF	ON	SET GLOBALS

With this parameter, you determine whether a page eject is to be performed as a result of a logical page break, a break between program input and output, and the "normal end" message.

EJ=ON	A page eject will be performed.
EJ=OFF	A page eject will not be performed. This may be used to save paper during test runs, where page ejections are not needed.

The EJ setting may be overridden using the EJECT statement.

The EJ parameter only applies to the first report (report 0). For additional reports, the EJECT statement with report specification has to be used.

Example:

```
SET GLOBALS EJ=OFF
```

EM - Edit Mask

Parameter	Possible Values	Default Value	Applicable Statements
EM	see below	none	DEFINE DATA DISPLAY FORMAT INPUT MOVE EDITED PRINT WRITE

With this parameter, you can specify an edit mask for an input and/or output field that is used in a DISPLAY, INPUT, MOVE EDITED, PRINT or WRITE statement.

For input fields, values must be entered exactly matching the edit mask. If it is desired to display the edit mask for an input field, the field should be defined as modifiable (AD=M).

For a database field, a default edit mask may have been defined in the DDM. If you specify with the EM parameter an edit mask for a database field, this edit mask specified will be used instead of any default edit mask which may be defined for the field in the DDM.

If you specify EM=OFF for a field, no edit mask will be used for the field, not even one that may be defined in the DDM.

At statement level of a DISPLAY, FORMAT, INPUT or WRITE statement, no detail field edit mask may be specified, except EM=OFF.

An edit mask overrides any settings for the session parameters AL, NL and SG.

Examples:

```
DISPLAY AA(EM=OFF) AB(EM=XX.XX) WRITE SALARY (EM=ZZZ,ZZ9)
```

An abbreviated notation can be used for long edit masks. The following examples demonstrate the abbreviated notation which may be used for numeric, hexadecimal, and alphanumeric edit masks:

EM=9(4)-9(5) is equivalent to: EM=9999-99999

EM=H(10) is equivalent to: EM=HHHHHHHHHHH

EM=X(6)..X(3) is equivalent to: EM=XXXXXX..XXX

This notation may be used only with the characters "9", "H", "X" and "Z" which represent significant print positions in the numeric, hexadecimal, and alphanumeric edit masks.

Blanks in Edit Masks

Blanks within an edit mask are represented by the character on your keyboard which in hexadecimal code corresponds to H'20' (ASCII) or H'5F' (EBCDIC) respectively, that is, the character "^" (or "¬").

Default Edit Masks

If no edit mask is specified for a field, a default edit mask is assigned to the field depending on the field format:

Field Format	Default Edit Mask
A	X
B	H
N, P, I	Z9
F	scientific representation
D	depends on default date format (as set with the profile parameter DTFORM)
T	HH:II:SS
L	blank / X

Edit Masks for Numeric Fields

An edit mask specified for a field of format N, P, I, or F must contain at least one "9" or "Z". If more 9s or Zs exist than the number of positions contained in the field value, the number of print positions in the edit mask will be adjusted to the number of digits defined for the field value. If fewer 9s or Zs exist, the high-order digits before the decimal point and/or low-order digits after the decimal point will be truncated.

Two options are available when using numeric edit masks. The first option allows the insertion of a special character as the first character in the edit mask.

Characters for the Definition of Numeric Edit Masks:

Character	Function
9	Position to be displayed (one digit of the field value).
Z	Zero suppression for leading zeros. This is the default for numeric fields. The letter "Z" may be repeatedly specified to represent floating zero suppression. A "Z" must not be specified to the right of the decimal point. A zero value may be displayed as blanks using all "Z"s in the edit mask (see also the session parameter ZP).
+	A floating sign is to be displayed preceding/following the number. The sign may be generated as a plus or minus depending on the value of the field.
-	A floating minus is to be displayed preceding/following the number if the value of the field is negative.
S	A sign is to be displayed to the left of the value. A plus sign is displayed for a positive value and a minus sign is displayed for a negative value. If "S" is used, it must be specified as the first character in an edit mask.
N	A minus sign is to be displayed to the left of the column if the value of the field is negative.
.	A period, if used as the first character, represents a decimal position and is significant.
H	Designates the beginning of a hexadecimal edit mask. If an "H" appears as the first position, all other character which are not "H" will be considered insertion characters.

The second option allows any number of leading characters to appear prior to the first displayable position (as indicated with the digit "9"). The first of these leading characters must not be any of the edit characters described above unless enclosed in quotes. The first of these insertion characters specified will appear in the output only if the value contains leading zeros and the edit mask is defined with Z (leading zero suppression). This character will then be displayed instead of a blank for leading zeros.

Both options permit the use of insertion and trailing characters. The symbol (^) may be used to represent a leading, inserted, or trailing blank. By enclosing significant characters (9, H, Z, X) in apostrophes, it is possible to use any characters as leading, insertion, or trailing characters. Insignificant edit mask characters need not be enclosed in apostrophes. Within the same edit mask notation, it is possible to have groups of leading, insertion, and/or trailing character strings, some of which are bounded by apostrophes and some of which are not.

A trailing sign character can be specified for numeric edit masks by using the "+" or "-" characters as the last character in the edit mask. A "+" will produce a trailing "+" or "-" sign depending on the value of the field. A "-" will produce a trailing space or "-" sign depending on the value of the field. If a leading and trailing sign are specified in the edit mask, both will be produced.

Examples of Numeric Edit Masks:

The table below lists the results obtained from the original values shown at the top of each column. All values used as column headings represent N format fields:

Value	0000.03 (n 4.2)	-0054 (n 4)	+0087 (n 4)	0962 (n 4)	1830 (n 4)
Edit Mask					
EM=9.9	0.0	4.	7.	2.	0.
EM=99	00	54	87	62	30
EM=S99	+00	-54	+87	+62	+30
EM=+Z9	+0	-54	+87	+62	+30
EM=-9.99	0.03	-4.	7.	2.	0.
EM=N9	0	-4	7	2	0
EM=*9.99	0.03	4.	7.	2.	0.
EM=Z99	00	54	87	962	830
EM=*DMZZ9.9	DM**0.0	DM*54.	DM*87.	DM962.	DM830.
EM=999+	000+	054-	087+	962+	830+
EM=999-	000	054-	087	962	830
IC=\$ EM=ZZZ.99	\$.03	\$54.	\$87.	\$962.	\$830.
EM=H(6)					
- ASCII:	303030303033	30303574	30303837	30393632	31383330
- EBCDIC:	F0F0F0F0F0F3	F0F0F5D4	F0F0F8F7	F0F9F6F2	F1F8F3F0

By combining edit masks with the parameters IC and TC, negative numbers can be displayed in varying formats using a DISPLAY statement.

Edit Masks for Alphanumeric Fields

An alphanumeric edit mask which is only to be used with A format fields must contain at least one "X" which represents a character to be displayed. An "H" as the first character designates a hexadecimal edit mask. A blank is represented by a (^) symbol. All other characters except parentheses are permissible including leading, trailing, and insertion characters. It is also possible to specify leading, insertion, or trailing characters enclosed within apostrophes. If the character "X", a closing parenthesis, or a quotation mark is specified as an insertion character, it must be enclosed within apostrophes.

If leading characters are used before the first displayable position X, the first of these leading characters will not be displayed. Trailing characters which immediately follow the last permissible print position will be displayed.

The number of positions to be displayed will be adjusted to the length of the edit mask if the mask is shorter than the field.

Example of Alphanumeric Edit Masks:

The following program lists the alphanumeric edit masks for a field that is defined with format/length A4 and contains the value "BLUE".

```
* EXAMPLE 'EMMAGE' DEFINE DATA LOCAL 1 EVERY {A} INIT <'BLANK'>
    DOO=DOORNO.AUTOTR=TOTALS(IMAGE $) IN RANGE (DOO=N,N,N) / IMAGE $() IN RANGE (DOO=STATISTICS) / IMAGE $() IN RANGE (DOO=DOO,N,N,N) / IMAGE $() IN RANGE (DOO=N,N,N,N,N,N) / IMAGE $() IN RANGE (DOO=DOO,DOO,DOO,DOO,DOO,DOO) / IMAGE $() IN RANGE (DOO=DOO,DOO,DOO,DOO,DOO,DOO,DOO,DOO) / IMAGE $() IN RANGE (DOO=DOO,DOO,DOO,DOO,DOO,DOO,DOO,DOO,DOO,DOO)
```

MASK 1:	B.L.U.E	MASK 2:	B L U E	MASK 3:	B--L--U	MASK 4:	B-L-U-E-	MASK 5:	B L U E	MASK 6:	BL...UE	MASK 7:	234BLUE
---------	---------	---------	---------	---------	---------	---------	----------	---------	---------	---------	---------	---------	---------

Hexadecimal Edit Masks

If the character "H" is specified as the first character in an edit mask, the content of an alphanumeric or numeric field will be displayed in hexadecimal format. Each "H" represents two print positions that will occur for each byte in the source field. Characters other than "H" serve as insertion or trailing characters in the mask. The number of positions to be displayed will be adjusted to the length of the edit mask if the mask is shorter than the field. The length of the edit mask will be adjusted to the length of the field if the field length is shorter than the edit mask.

Insertion or trailing characters may be optionally specified bounded by apostrophes.

All fields displayed with a hexadecimal edit mask are treated as alphanumeric. Therefore, if the edit mask is shorter than the field to be edited, numeric or alphanumeric positions will be displayed from left to right disregarding any decimal point positions.

Edit Mask Examples for Hexadecimal Fields:

The tables below list the hexadecimal edit masks with results obtained from the original fields and values shown above each column. All numeric values ("-10", "+10", "01") to which edit masks have been applied originated in fields defined with N2 format. The alphanumeric value "AB" originated from a field defined with format/length A2.

ASCII:

Value ->	AB	-10	+10	01
EM=HH	4142	3170	3130	3031
EM=H^H	41 42	31 70	31 30	30 31
EM=HH^H	4142	3170	3130	3031
EM=H-H	41-42	31-70	31-30	30-31
EM=H	41	31	31	30

Note:

In the case of em=h(n) (Hexadezimal-Output) the value of n must be in the range of $1 \leq n \leq 126$. Is $n > 126$, then an error will be displayed if the variable has more than 126 elements.

Example:

```
A (A100) := 'A' /* 100 characters/elements PRINT A (EM=H(200)) /* no errorr, as 100<=126, even if 200>126 END
```

EBCDIC:

Value ->	AB	-10	+10	01
EM=HH	C1C2	F1D0	F1F0	F0F1
EM=H:H	C1 C2	F1 D0	F1 F0	F0 F1
EM=HH:H	C1C2	F1D0	F1F0	F0F1
EM=H-H	C1-C2	F1-D0	F1-F0	F0-F1
EM=H	C1	F1	F1	F0

Edit Masks for Date and Time Fields - Formats D and T

In edit masks for fields which are defined with format D (date) or T (time), the following characters can be specified:

Date - format D, and Time - format T:

Character	Usage
DD	Day.
ZD	Day, with zero suppression.
MM	Month.
ZM	Month, with zero suppression.
YYYY	Year, 4 digits.
YY	Year, 2 digits. (If used for input fields, the current century is concatenated before the input value.)
Y	Year, 1 digit. Must not be used for input fields.
WW	Number of Week.
ZW	Number of Week, with zero suppression.
JJJ	Julian day.
ZZJ	Julian day with zero suppression.
NN... or N(n)	Name of Day (language-dependent). The maximum length is determined by the number of N's or by n. If the name is longer than the maximum length, it will be truncated; if it is shorter, the actual length of the name will be used.
O	Number of week day (Monday = 1, Tuesday = 2, etc. The profile parameter DTFORM, as described in your Natural Operations documentation, determines whether Monday or Sunday is considered the first day of the week).
LL... or L(n)	Name of Month (language-dependent). The maximum length is determined by the number of L's or by n. If the name is longer than the maximum length, it will be truncated; if it is shorter, the actual length of the name will be used.
R	Year in Roman numerals (maximum 13 digits).

If only year (YY or YYYY) but no month or day is specified within an input edit mask, the values for month and day will both be set to "01". If only year (YY or YYYY) and month (MM) but no day is specified within an input edit mask, the value for day will be set to "01".

If a week is displayed which belongs to the previous or the next year, the display for year is modified accordingly.

Whether a week is week 53 of the old year or week 1 of the new year, depends on which year the Thursday of that week belongs to: if the Thursday is in the old year, the week belongs to the old year; if it is in the new year, the week belongs to the new year.

Note:

"MM" (or "ZM") and "LL" (or "L(n)") must not be specified together in an edit mask. "NN" (or "N(n)") and "O" must not be specified together in an edit mask.

Time - format T - only:

Character	Usage
T	Tenths of a second.
SS	Seconds.
ZS	Seconds, with zero suppression.
II	Minutes.
ZI	Minutes, with zero suppression.
HH	Hours.
ZH	Hours, with zero suppression.
AP	AM/PM element.

Examples of Date and Time Edit Masks:

[illegible]

Edit Masks for Logical Fields - Format L

For fields of format L (logical fields), edit masks can be defined as follows:

$$(EM = [false-string/] true-string)$$

The false-string must not be longer than 31 characters.

Example of Edit Masks for Logical Field:

TRUE	INDEX =	1	ON	INDEX =	1	FALSE	INDEX =	2	OFF	INDEX =	2	TRUE	INDEX =	3	ON	INDEX =	3	FALSE	INDEX =	4	OFF	INDEX =	4	TRUE	INDEX =	5	ON	INDEX =	5
------	---------	---	----	---------	---	-------	---------	---	-----	---------	---	------	---------	---	----	---------	---	-------	---------	---	-----	---------	---	------	---------	---	----	---------	---

ES - Empty Line Suppression

Parameter	Possible Values	Default Value	Applicable Statements
ES	ON / OFF	OFF	DISPLAY FORMAT WRITE

With this parameter, you can suppress the printing of empty lines generated by a DISPLAY or WRITE statement.

If ES=ON is specified, a line resulting from a DISPLAY or WRITE statement which contains all blank values will not be printed.

ES=ON is particularly useful when displaying arrays (for example, multiple-value fields or fields contained within a periodic group) to avoid printing a large number of empty lines.

To achieve empty suppression for numeric values, the field must be specified with ZP=OFF and ES=ON in order to have null values printed as blanks.

See also the session parameters IS and ZP.

Example:

```
DISPLAY (ES=ON) NAME CITY
```

FC - Filler Character

Parameter	Possible Values	Default Value	Applicable Statements
FC	any character	blank	DISPLAY FORMAT

With this parameter, you specify the filler character which will appear on either side of a heading produced by a DISPLAY statement across the full column width. FC only applies if the column width is determined by the field length and not by the header (see also session parameter HW); otherwise FC will be ignored.

Example:

```
DISPLAY (FC=*)
```

FCDP - Filler Character for Dynamically Protected Fields

Parameter	Possible Values	Default Value	Applicable Statements
FCDP	ON / OFF	ON	SET GLOBALS

With this parameter, you can suppress the display of filler characters for input fields that have been made write-protected dynamically (that is, to which the attribute AD=P has been assigned via a control variable).

Depending on the setting of this parameter, dynamically protected input fields are displayed filled either with blanks or with the defined filler characters:

FCDP=ON	Dynamically protected input fields are displayed filled with filler characters. This may suggest to the users that they could enter something in the fields.
FCDP=OFF	Dynamically protected input fields are displayed filled with blanks.

Example:

```
DEFINE DATA LOCAL 1 #FIELD1 (A5) 1 #FIELD2 (A5) 1 #CVAR1 (C) INIT <(AD=P)> 1 #CVAR2 (C) END-DEFINE * INPUT #FIELD1 (AD=Y...' CV=#CVAR1) /* field is protected      #FIELD2 (AD=Y...' CV=#CVAR2) /* field is not protected ... END
```

Display with FCDP=ON: #FIELD1 _____ #FIELD2 _____

Display with FCDP=OFF: #FIELD1 #FIELD2 _____

FL - Floating Point Mantissa Length

Parameter	Possible Values	Default Value	Applicable Statements
FL	1 to 16	none	DISPLAY FORMAT INPUT PRINT WRITE

With this parameter, you specify the mantissa length of a floating point variable during input or output.

The total length is "FL + 6" for sign, exponent, and decimal character.

```
DISPLAY FL=5 -> +1.2345E+03
```

FS - Format Specification

Parameter	Possible Values	Default Value	Applicable Statements
FS	ON / OFF	OFF	SET GLOBALS

This parameter applies to reporting mode only.

This parameter indicates whether a default format/length setting is to be in effect for the definition of user-defined variables.

FS=OFF	A user-defined variable in a Natural program for which no format/length is specified will be assigned the default format/length N7.
FS=ON	No default format/length will be assigned by Natural for a newly introduced variable in reporting mode. Format/length must be explicitly specified.

Example:

```
SET GLOBALS FS=ON
```

GC - Filler Character for Group Headers

Parameter	Possible Values	Default Value	Applicable Statements
GC	any character	blank	DISPLAY FORMAT

With this parameter, you specify the filler character which will appear on either side of a group heading produced by a DISPLAY statement across all field columns that belong to that group. Unlike the FC parameter, which applies to individual columns, the GC parameter applies to headings across a group of columns.

Example:

```
DISPLAY (GC=*)
```

HC - Header Centering

Parameter	Possible Values	Default Value	Applicable Statements
HC	C, L, R	C	DISPLAY FORMAT

This parameter determines the placement of column headers.

HC=C	Headers will be centered.
HC=L	Headers will be left-justified.
HC=R	Headers will be right-justified.

Example:

```
DISPLAY (HC=L)
```

HE - Helproutine

Parameter	Possible Values	Default Value	Applicable Statements
HE	see below	none	INPUT

With this parameter, you specify the name of a helproutine or map which is to be assigned to a field.

Helproutines can be created with the Natural program editor, help maps with the Natural map editor.

The helproutine may then be invoked during processing of an INPUT statement or a map by positioning the cursor under the field and pressing the help function key (as defined with the SET KEY statement) or by entering the help character ("?" by default) into the field.

The syntax of this parameter is:

$$HE = operand1 \left[\left\{ \begin{array}{c} operand2 \\ = \end{array} \right\} \right] \dots 20$$

Operand	Possible Structure				Possible Formats												Referencing Permitted	Dynamic Definition
Operand1	C	S			A												no	no
Operand2	C	S	A		A	N	P	I	F	B	D	T	L	C	G	O	no	no

Operand1 is the name of the helproutine or map to be invoked. The name may be a 1- to 8-character alphanumeric constant or user-defined variable. If a variable is used, it must have been previously defined. The name may contain an ampersand (&); at execution time, this character will be replaced by the current value of the Natural system variable *LANGUAGE. This feature allows the use of multi-lingual helproutines or maps.

After the helproutine or map name, you may specify 1 to 20 parameters (operand2) which may be passed to the helproutine or map. The parameters may be specified as constants or as user-defined variables. If an "=" is specified as a parameter, the name of the field as defined in the map definition will be passed to the helproutine; in the case of a helproutine which is assigned to a map, "=" denotes the name of the map.

Note:

The operands must be separated either by the input delimiter character (as specified with the session parameter ID) or by a comma. A comma must not be used for this purpose, however, if the comma is defined as decimal character (with the session parameter DC).

If parameters are specified, the helproutine must begin with a DEFINE DATA PARAMETER statement which defines fields that correspond in format and length with the parameters.

If the parameter notation "=" is used to pass a field or map name, the corresponding parameter in the helproutine must be specified as A65.

The value of the field for which a helproutine is specified may be referenced within the helproutine. This is done by specifying a field in the DEFINE DATA PARAMETER statement which corresponds in format and length with the original field. In the block of fields defined within the DEFINE DATA PARAMETER statement, this field must always be defined behind the parameters, if present.

If the field for which a help routine is specified is an array element, its indices may be referenced by the help routine. To do so, you specify index parameters with format/length I2 at the end of the DEFINE DATA PARAMETER statement. You may specify up to three index parameters according to array dimensions.

Execution Of Help routines

If a help routine is requested - by entering a "?" in the field, or by pressing the help key (as defined with a SET KEY statement), or via a REINPUT USING HELP statement - all other data that may have been entered into fields are not assigned to the program variables until all help requests have been processed.

Note:

Only one help request per INPUT statement is possible; that is, if help is requested for more than one field (for example, by entering question marks in multiple fields), only the first help request will be executed.

Example:

```
/* MAIN PROGRAM DEFINE DATA 1 #A(A20/1:3)
                                END-DEFINE ... SET KEY PF1=HELP
                                ... INPUT #A (2) (HE='HELPA', =) ...
                                END
```

```
/* HELP-ROUTINE 'HELPA' DEFINE DATA PARAMETER 1 #VARNAME (A65) 1 #PARM1 (A20) 1 #VARINDEX (I2) END-DEFINE ...
```

If a value is to be passed from a help routine to an input field, the field must be defined as modifiable (AD=M).

HW - Heading Width

Parameter	Possible Values	Default Value	Applicable Statements
HW	ON / OFF	ON	DISPLAY FORMAT

With this parameter you determine the width of a column output with a DISPLAY statement.

HW=ON	The width of a DISPLAY column is determined by either the length of the heading text or the length of the field, whichever is longer. This is true even if no heading text is output, either because the DISPLAY statement contains the keyword NOHDR or the DISPLAY statement is a subsequent DISPLAY (see also the DISPLAY statement).
HW=OFF	The width of a DISPLAY column is determined by the length of the field. HW=OFF only applies to DISPLAY statements which do not create headers (that is, either a first DISPLAY statement with NOHDR option or a subsequent DISPLAY statement).

Example:

```
DISPLAY (HW=OFF)
```

IA - INPUT Assign Character

Parameter	Possible Values	Default Value	Applicable Statements
IA	any special character	=	SET GLOBALS

With this parameter, you determine the character to be used as the assignment character for the input parameter processing in INPUT statements in keyword/delimiter mode, or when processing data from the data stack.

The character specified with the IA parameter must not be the same as the character specified with the CF parameter (control character for terminal commands), DC parameter (decimal character) or ID parameter (input delimiter character).

Note:

The character specified with the IA parameter should not be the same as the one specified with the HI parameter (help character).

Example:

```
* Program 'SAMPLE' DEFINE DATA LOCAL 1 #A (A1) 1 #B (A1) END-DEFINE INPUT #A #B WRITE #A #B END
```

Enter the system command **GLOBALS IM=D**

Then enter the system command **SAMPLE #A=Y,#B=X**

The program produces the following output: Y X

Enter the system command **GLOBALS IA=:**

Then enter the system command **SAMPLE #B:X,#A:Y**

The program produces the following output: Y X

IC - Insertion Character

Parameter	Possible Values	Default Value	Applicable Statements
IC	any character	none	DISPLAY FORMAT

With this parameter, you specify the character string to be inserted in the column immediately preceding the value of a field output with a DISPLAY statement. The width of the output column is enlarged accordingly. You can specify a string of one to ten characters.

For numeric values, the insertion characters will be placed before the first significant digit printed.

Insertion characters may optionally be specified within apostrophes, in which case any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. A blank in a character string not enclosed within apostrophes is represented by the character "^".

The IC and LC parameters are mutually exclusive.

Examples:

```
DISPLAY AA(IC=*) DISPLAY SALARY(IC=' $ ')
```

ID - INPUT Delimiter Character

Parameter	Possible Values	Default Value	Applicable Statements
ID	any special character, blank	, (comma)	SET GLOBALS

With this parameter, you specify the character to be used as a delimiter for INPUT statements in keyword/delimiter mode.

If the ID setting is to be a comma, it must be specified as ID=', ' when using the dynamic parameter facility, because the character "," separates individual parameters.

The character specified with this parameter must not be the same as the one specified with the DC parameter (decimal character) or IA parameter (input assign character).

Note:

The character specified with the ID parameter should not be the same as the one specified with the CF parameter (control character for terminal commands) or HI parameter (help character).

The period "." should not be used as input delimiter either, because this might lead to situations in which a program termination period would be misinterpreted as input delimiter.

Examples:

```
SET GLOBALS ID=/ PARM= ' . . . , ID= ' , ' , . . . '
```

IM - INPUT Mode

Parameter	Possible Values	Default Value	Applicable Statements
IM	F, D	F	SET GLOBALS

With this parameter, you determine the input mode.

IM=D	Delimiter mode.
IM=F	Forms mode.

The default mode for video terminal input is forms mode.

The IM setting may also be changed with the terminal commands %D and %F.

See the INPUT statement for additional information on delimiter mode and forms mode.

Example:

```
SET GLOBALS IM=D
```

IP - INPUT Prompting Text

Parameter	Possible Values	Default Value	Applicable Statements
IP	ON / OFF	ON	FORMAT INPUT

This parameter is used to control prompting text in INPUT statements.

IP=ON	Even if no text is specified preceding the input/output in an INPUT statement, the name of the field will be generated by default as a text element preceding the field as prompting text.
IP=OFF	No automatic prompting text will be generated for input/output fields in an INPUT statement. Only fields explicitly preceded with a text element will receive the text as prompting text.

Example:

```
FORMAT IP=OFF
```

IS - Identical Suppress

Parameter	Possible Values	Default Value	Applicable Statements
IS	ON / OFF	OFF	DISPLAY FORMAT WRITE

With this parameter, you can suppress the printing of identical information in successive lines created by WRITE or DISPLAY statement.

If IS=ON is specified, a value which is identical to the previous value for the field will not be displayed.

If a DISPLAY or WRITE statement is used to create multiple output lines using the VERT or "/" notation, IS=ON applies only to the first line.

The IS parameter setting can be suspended for one record by issuing the SUSPEND IDENTICAL SUPPRESS statement.

The IS parameter may be used in combination with the parameters ES and ZP to cause empty line suppression.

Example:

```
FORMAT IS=ON
```

KD - Key Definition

Parameter	Possible Values	Default Value	Applicable Statements
KD	ON / OFF	OFF	FORMAT

This parameter is used to display the names assigned to the PF keys (see the SET KEY statement).

This information will always be displayed automatically in the two bottom lines of the physical screen with any output created by the INPUT, WRITE, DISPLAY, and PRINT statement.

As this display requires two lines, the logical page size (see the session parameter PS) must be reduced by two.

Note for graphical user interfaces:

If PF keys are defined, they are always displayed, regardless of the setting of this parameter. If no PF keys are defined, this parameter can be used to switch on/off the display of the ENTER key.

Example:

```
FORMAT KD=ON
```

LC - Leading Characters

Parameter	Possible Values	Default Value	Applicable Statements
LC	any character	none	DISPLAY FORMAT

With this parameter, you can specify leading characters are displayed immediately before a field output by a DISPLAY statement. The width of the output column is enlarged accordingly.

Up to 10 characters may be specified. Leading characters may optionally be specified enclosed within apostrophes, in which case, any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes. The character "^" is used to represent a blank in a character string not enclosed within apostrophes.

The session parameters LC and IC are mutually exclusive.

Example:

```
DISPLAY LC=*
```

LE - Limit Error Processing

Parameter	Possible Values	Default Value	Applicable Statements
LE	ON / OFF	OFF	SET GLOBALS

With this parameter, you determine the action to be taken if the limit specified for processing loop execution is reached.

The limit may be either a global limit or a limit specified for a given processing loop.

LE=OFF	The Natural program will be terminated normally at the point where the limit was reached.
LE=ON	The Natural program will be terminated and an error message will be issued when the limit is reached.

Example:

```
SET GLOBALS LE=OFF
```

LS - Line Size

Parameter	Possible Values	Default Value	Applicable Statements
LS	2 to 250 (if a default page title is used, the minimum value is 8)	physical line size	DISPLAY FORMAT INPUT SET GLOBALS WRITE

With this parameter, you determine the maximum number of characters permitted per line for DISPLAY, INPUT and WRITE statements.

Example:

```
DISPLAY LS=100
```

LT - Limit of Records Read

Parameter	Possible Values	Default Value	Applicable Statements
LT	0 to <i>n</i> (<i>n</i> = value of <i>profile</i> parameter LT at session start)	99999999	SET GLOBALS

With this parameter, you determine the maximum number of records which may be read in any given processing loop in a Natural program.

This limit applies to any processing loop initiated with a READ, FIND or HISTOGRAM statement. All records read, including rejected records from a WHERE clause, are counted against this limit.

If a limit for a single processing loop (as set with a LIMIT statement or a limit notation) is higher than the limit set with the LT parameter, the limit set with the LT parameter applies.

See also the session parameter LE.

Example:

```
SET GLOBALS LT=100
```

MC - Multiple-Value Field Count

Parameter	Possible Values	Default Value	Applicable Statements
MC	1 to 191	1	DISPLAY FORMAT INPUT PRINT WRITE

This parameter may be used in reporting mode only.

With this parameter, you determine the number of values of a multiple-value field to be output by default when the field is specified without an index in a DISPLAY or WRITE statement.

Example:

```
FORMAT MC=5
```

MP - Maximum Number of Pages

Parameter	Possible Values	Default Value	Applicable Statements
MP	1 to 99999	32767	DISPLAY FORMAT PRINT WRITE

With this parameter, you determine the maximum number of pages to be produced for a report.

The value specified is the number of physical pages and has no effect on the starting page number used.

The program will be terminated with an error message if the MP value is exceeded.

Example:

```
FORMAT MP=1000
```

MS - Manual Skip

Parameter	Possible Values	Default Value	Applicable Statements
MS	ON / OFF	OFF	FORMAT INPUT

With this parameter, you control the cursor positioning during the processing of an INPUT statement.

If MS=OFF is specified, the cursor will be positioned to the next input field as soon as the value for the current field is entered with all positions.

Example:

INPUT (MS=ON) #A #B

Note: The setting MS=ON is not supported under BS2000/OSD.

MT - Maximum CPU Time

Parameter	Possible Values	Default Value	Applicable Statements
MT	see below	60 (seconds)	SET GLOBALS

This parameter only applies to programs executing on mainframe computers in batch mode. The limit for programs operating in interactive mode is controlled by the TP monitor in use.

With this parameter, you determine the maximum amount of CPU time which may be used by a Natural program. The value is specified in seconds.

MT=0 indicates that no CPU time limit is to be in effect.

The maximum value that can be used is determined by the operating system environment. Any value greater than the maximum will be reduced to the maximum supported by the operating system.

In system environments which do not support CPU time measurement, the limit is interpreted as elapsed time. In systems without timer support, the CPU time limit is ignored.

Example:

SET GLOBALS MT=5

NC - Use of Natural System Commands

Parameter	Possible Values	Default Value	Applicable Statements
NC	ON / OFF	OFF	SET GLOBALS

With this parameter, you determine whether Natural system commands can be used during the Natural session or not.

NC=OFF	All system commands can be used.
NC=ON	System commands cannot be used - except FIN, LAST, LOGOFF, LOGON, MAINMENU, RENUMBER, RETURN, SETUP, and TECH.

Note:

Natural terminal commands and user-created commands are not affected by this parameter.

Example:

```
GLOBAL$ NC=OFF
```

NL - Numeric Length for Output

Parameter	Possible Values	Default Value	Applicable Statements
NL	see below	none	DISPLAY FORMAT INPUT PRINT WRITE

This parameter determines the default input/output length for a numeric field used in a DISPLAY, INPUT, PRINT or WRITE statement.

The length is specified as nn.m, where nn represents the number of positions before the decimal point and m represents the number of positions after the decimal point. The m notation is optional. The value of m must not exceed 7. The total of n+m must not exceed 29.

If NL is set less than the field length, values are truncated. No error is produced when relevant digits are truncated.

If NL is set greater than the field length, values are expanded with blanks. No error is produced when an input field is truncated.

The NL parameter must not be specified for groups.

Any edit mask specified for a field will override the NL parameter for this field.

Example:

```
DISPLAY #AA(NL=20) #AB(NL=3.2)
```

OPF - Overwriting of Protected Fields by Help routines

Parameter	Possible Values	Default Value	Applicable Statements
OPF	ON / OFF	ON	SET GLOBALS

This parameter determines whether the content of a write-protected field (attribute AD=P) can be overwritten by a help routine assigned to the field.

OPF=ON	A help routine assigned to a field can overwrite the field's content, even if the field is write-protected.
OPF=OFF	Help routines cannot overwrite the contents of write-protected fields.

The OPF parameter only applies to the field for which a help routine is invoked, it does not affect parameters explicitly passed to the help routine. This means that the OPF parameter takes no effect if the field for which help is invoked is also explicitly specified as a parameter to be passed to the help routine.

PC - Periodic Group Count

Parameter	Possible Values	Default Value	Applicable Statements
PC	1 to 99	1	DISPLAY FORMAT INPUT PRINT WRITE

This parameter may be used in reporting mode only.

This parameter determines the number of periodic group occurrences to be output by default if a periodic group (or a field contained within a periodic group) is specified without an index in a DISPLAY or WRITE statement.

Example:

```
FORMAT PC=5
```

PD - NATPAGE Page Dataset

Parameter	Possible Values	Default Value	Applicable Statements
PD	0 to 255	50	SET GLOBALS

This parameter is not available under Windows 98 and Windows NT/2000.

This parameter is used to specify the maximum number of pages (screens) which can be recorded with the NATPAGE utility.

If this number is exceeded, each additional screen will overwrite an already recorded screen, thus gradually overwriting the oldest screens, i.e. those recorded first (wrap-around technique).

The recorded screens are stored in the Natural system file FUSER.

For further information on the NATPAGE utility, see the terminal commands %E, %I, %O, %P and %S.

Example:

```
SET GLOBALS PD=60
```

PM - Print Mode

Parameter	Possible Values	Default Value	Applicable Statements
PM	C, D, I, N	none	DEFINE DATA DISPLAY FORMAT INPUT PRINT SET GLOBALS* WRITE

* With the SET GLOBALS statement, the Natural profile parameter PM (as described in your Natural Operations documentation) can be set, not the session parameter PM.

This parameter is used to indicate how fields are to be displayed.

PM=C	(can only be set on mainframe computers) An alternative character set is used (see the module NATPM in the Natural source library).
PM=D	(can only be set on mainframe computers) Defines DBCS fields without shift codes (see Support of Double-Byte Character Sets in the Natural Operations for Mainframes documentation).
PM=I	Field values are displayed in inverse direction; that is, from right to left (for example, for use in Middle East countries).
PM=N	No hardcopy of the display can be made.

Example:

```
LIMIT 1 READ EMPLOYEES DISPLAY NOTITLE NAME DISPLAY NOTITLE NAME (PM=I) DISPLAY NOTITLE NAME END
```

NAME ----- MORENO	ONEROM MORENO
-------------------	---------------

Note:

More than one value may be specified.

PS - Page Size

Parameter	Possible Values	Default Value	Applicable Statements
PS	10 to 250, 0 (see below)	physical page size	DISPLAY FORMAT INPUT SET GLOBALS WRITE

This parameter determines the maximum number of lines per page to be used for Natural reports created with the DISPLAY/WRITE statement.

If PS=0 is specified for the first report to be output (report 0), the physical device page size minus 1 will be used. If PS=0 is specified for reports 1 - 31, this will cause automatic newpage processing to be inhibited, that is, no automatic page break processing will be performed.

PS=0 can only be specified with a FORMAT statement.

Example:

```
FORMAT PS=40
```

REINP - Internal REINPUT for Invalid Data

Parameter	Possible Values	Default Value	Applicable Statements
REINP	ON / OFF	ON	SET GLOBALS

By default, Natural automatically issues an internal REINPUT statement if invalid data have been entered in response to an INPUT statement. With this parameter, you can switch this mechanism off. This will allow you to handle such input errors yourself in your application.

REINP=ON	An internal REINPUT statement is issued when invalid data have been entered.
REINP=OFF	An internal REINPUT statement is not issued when invalid data have been entered.

SA - Sound Terminal Alarm

Parameter	Possible Values	Default Value	Applicable Statements
SA	ON / OFF	OFF	SET GLOBALS

This parameter determines whether the terminal alarm feature is to be used.

The use of this feature requires that the terminal alarm hardware feature has been installed for the terminal, otherwise the parameter will be ignored.

SA=ON	The terminal alarm will be sounded each time the user is prompted for input by Natural.
SA=OFF	The terminal alarm feature will not be used when the user is prompted for input. However, the alarm may still be activated with the SOUND ALARM option of the REINPUT statement.

Example:

SET GLOBALS SA=ON

SF - Spacing Factor

Parameter	Possible Values	Default Value	Applicable Statements
SF	1 to 30	1	DISPLAY FORMAT SET GLOBALS

This parameter determines the default number of spaces to be inserted between columns field values on Natural reports created with the DISPLAY statement.

The SF parameter cannot be set to "0"; that is, at least one blank character must be placed between report columns.

Example:

GLOBALs SF=5

SG - Sign Position

Parameter	Possible Values	Default Value	Applicable Statements
SG	ON / OFF	ON	DISPLAY FORMAT INPUT PRINT WRITE

This parameter determines whether or not a sign position is to be allocated for a numeric field.

SG=OFF causes numeric fields with negative values to be output without a "-" sign.

Example:

```
FORMAT SG=OFF
```

If the EM parameter is specified, it overrides the SG parameter.

SL - Source Line Length

Parameter	Possible Values	Default Value	Applicable Statements
SL	20 to 80	72	GLOBALS command

This parameter only applies to Natural on mainframe computers.

This parameter determines the number of characters to be interpreted on each Natural source-code line; this only applies to EDT mode (which is activated with the system command EDT).

Example:

```
GLOBALS SL=74
```

SM - Structured Mode

Parameter	Possible Values	Default Value	Applicable Statements
SM	ON / OFF	OFF	GLOBALS command

This parameter determines whether structured mode has to be used.

If structured mode (SM=ON) was specified by the Natural administrator during Natural installation, this parameter cannot be set to "OFF". Otherwise with SM=OFF the mode may be changed as desired.

Example:

```
GLOBAL$ SM=ON
```

Note:

If Natural Security is installed, this parameter may be disabled by Natural SECURITY to the effect that structured mode is invariably in effect for a given library.

TC - Trailing Characters

Parameter	Possible Values	Default Value	Applicable Statements
TC	any character	none	DISPLAY FORMAT

With this parameter, you can specify trailing characters that are to be displayed immediately to the right of a field output with a DISPLAY statement. The width of the output column is enlarged accordingly.

Up to 10 characters may be specified. Trailing characters may optionally be specified enclosed within apostrophes, in which case any characters can be specified. Any character string specified which contains a closing parenthesis or a quotation mark must be enclosed within apostrophes.

Examples:

```
FORMAT TC=* DISPLAY (TC='*B*')
```

TS - Translate Output from Programs in System Libraries

Parameter	Possible Values	Default Value	Applicable Statements
TS	ON / OFF	OFF	GLOBALS command

This parameter only applies to Natural on mainframe computers.

This parameter is used to translate output from Natural system libraries (that is, libraries whose names begin with "SYS") using a translate table. This may be necessary for locations which have non-standard lower-case usage (for example, Middle East countries); see also session parameter PM.

Note:

If TS=ON, the profile parameter LC=OFF and the session parameter AD=T, both of which translate input to upper case, are ignored, as they would cause undesired character translation for special character sets.

UC - Underlining Character

Parameter	Possible Values	Default Value	Applicable Statements
UC	any character, OFF	- (hyphen)	DISPLAY FORMAT

This parameter determines the character that to be used as underlining character for the following:

- column headings generated by DISPLAY statements;
- page titles/trailers produced by WRITE TITLE/TRAILER statements with UNDERLINED option.

If you do not wish column headers to be underlined, you have two options:

- "UC= " - A blank line will be output instead of underlining.
- UC=OFF - The field values will be output immediately below the heading line, without any blank line in between.

You can specify UC=OFF only at the statement level of a DISPLAY statement; in this case, you cannot make any other UC specifications for individual fields in that statement.

Examples:

```
FORMAT UC=*DISPLAY (UC= ) NAME AGE (UC=+)
```

WH - Wait for Record in Hold Status

Parameter	Possible Values	Default Value	Applicable Statements
WH	ON / OFF	OFF	SET GLOBALS

This parameter only applies to Adabas databases.

This parameter determines the action to be taken if a required record is not available.

WH=OFF	An error message will be returned if any of these records cannot be placed in hold status.
WH=ON	The user will be placed in wait status until either the requested record becomes available, or an error message is issued due to the database system exceeding a time limit or other limit while attempting to place the record in hold status.

See the Natural Programming Guide for further information on hold record processing.

Example:

SET GLOBALS WH=ON

ZD - Zero Division Check

Parameter	Possible Values	Default Value	Applicable Statements
ZD	ON / OFF	ON	SET GLOBALS

This parameter determines the action to be taken if an attempt is made to perform a division operation in which the divisor is "0".

ZD=ON	An error message will result if an attempt to divide by "0" is detected by Natural.
ZD=OFF	Natural will return a result of "0" for any division operation in which the divisor was "0".

Example:

SET GLOBALS ZD=OFF

ZP - Zero Printing

Parameter	Possible Values	Default Value	Applicable Statements
ZP	ON / OFF	ON	DISPLAY FORMAT INPUT PRINT REINPUT SET GLOBALS WRITE

With this parameter you can suppress the display of a numeric field (format N, I, P or B) or time field (format T) which contains a value of all zeros.

ZP=ON	A field value which consists of all zeros will be displayed as one zero (for numeric field) or all zeros (for a time field) respectively.
ZP=OFF	A field value which consists of all zeros will not be displayed.

Example:

```
FORMAT ZP=OFF
```

```
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
#include '/FS/fs1027/cd_prod/wml/wml_includes/footer' #include '/FS/fs1027/cd_prod/wml/wml_includes/footer'
```